ZLW
AF

OIPE IAP35
NOV 14 2005
PATENT & TRADEMARK OFFICE

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Toan D. TRAN et al

Art Unit: 2661

Application No: 09/847,079

Examiner:
Ian N. Moore

Filed: May 1, 2001

For: BACK PRESSURE CONTROL SYSTEM FOR
NETWORK SWITCH PORT

TRANSMITTAL OF BRIEF ON BEHALF OF APPELLANT

COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

Notice of Appeal was filed in this case on October 5, 2005.

Submitted herewith in triplicate is Appellant's Brief. A check in the amount of $250 for the under 37 CFR 41.20(b)(2)(small entity) is also submitted herewith.

Respectfully submitted,

Daniel J. Bedell
Reg. No. 30,156

SMITH-HILL & BEDELL, P.C.
16100 N.W. Cornell Road, Suite 220
Beaverton, Oregon 97006

Tel. (503) 574-3100
Fax (503) 574-3197
Docket: ZETT 2151
Postcard: 11/05-18

Certificate of Mailing

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on the 10 day of November, 2005.

Penelope Stockwell

OIPE IAP35 NOV 14 2005 PATENT & TRADEMARK OFFICE

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Toan D. TRAN et al

Art Unit: 2661

Application No: 09/847,079

Examiner: Ian N. Moore

Filed: May 1, 2001

For: BACK PRESSURE CONTROL SYSTEM FOR NETWORK SWITCH PORT

BRIEF ON BEHALF OF APPELLANT

COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

REAL PARTY IN INTEREST

Integrated Device Technology, Inc.

RELATED APPEALS AND INTERFERENCES

None

STATUS OF CLAIMS

Claims 1-13 are pending.

Claims 1-4, 6, 7, 9 and 10 are rejected.

Claims 5, 8, and 11-13 are objected to as dependant on rejected claims.

No claims have been withdrawn.

STATUS OF AMENDMENTS

No amendment was filed subsequent to final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

The invention, as recited in independent claims 1, 6 and 9, is summarized as follows.

11/15/2005 HTECKLU1 00000034 09847079

01 FC:2401 250.00 OP

Claim 1

The invention as recited in claim 1 is an apparatus (FIG. 1, dev.10) for receiving and storing incoming cells derived from data transmissions conveyed on a network (FIG. 1, dev.12), and for then forwarding the cells elsewhere in the network. The claimed apparatus comprising:

a cell memory (FIG. 3, dev. 32) for sequentially receiving and storing the incoming cells in separate memory blocks, each of the memory blocks being identified by a unique identification number (BLOCK_ID) (para. 30, lines 4-10), and for thereafter reading out a cell stored in any one of the memory blocks when its BLOCK_ID is transmitted to the cell memory (para. 36, 7-11);

queuing means (FIG. 3, dev. 36) for sequentially generating BLOCK_IDs of memory blocks storing cells to be read out of the cell memory (para. 36, lines 1-7); and

memory control means (FIG. 3, dev. 30) for maintaining a BLOCK_ID queue (FIG. 3, dev. 40/41), for adding BLOCK_IDs generated by the queuing means to the BLOCK_ID queue in an order in which they are generated by the queuing means (para. 36. lines 1-7), and for removing BLOCK_IDs from the BLOCK_ID queue and transmitting them to the cell memory in an order in which the BLOCK_IDs were added to the BLOCK_ID queue such that the cell memory reads out the cells stored in the memory blocks identified by the BLOCK_IDs transmitted by the memory control means (para. 36. lines 1-7).

Claim 6

The invention as recited in claim 6 is a method for receiving and storing cells derived from data transmissions conveyed on a network (FIG. 1, dev.12, and for then forwarding the cells elsewhere in the network. The method comprises the steps of:

a. sequentially receiving and writing the cells into separate blocks of a cell memory, wherein each memory block is identified by a unique identification number (BLOCK_ID) (FIG. 3, dev. 32) (para. 30, lines 4-10);

b. generating a sequence of BLOCK_IDs of memory blocks currently storing cells (FIG. 3, dev. 36) (para. 36, lines 1-7);

c. adding each generated BLOCK_ID to a BLOCK_ID queue (FIG. 3, dev. 40/41) (para. 36. lines 1-7); and

d. successively removing each BLOCK_ID from the BLOCK_ID queue in an order in which BLOCK_IDs were added to the BLOCK_ID queue (FIG. 3, dev. 36) (para. 36. lines 1-11) whenever the BLOCK_ID queue contains BLOCK_IDs and first back pressure data indicates that BLOCK_IDs may be removed from the BLOCK_ID queue (para. 39, lines 1-6), and refraining from removing BLOCK_IDs from the BLOCK_ID queue when the first back pressure data indicates that BLOCK_IDs may not be removed from the BLOCK_ID queue (para 39, lines 7-11),

e. reading a cell out of the cell memory whenever the BLOCK_ID of the memory block in which it is stored is removed from the BLOCK_ID queue at step d (para. 36, lines 1-7).

Claim 9

The invention as recited in claim 9 is a method for receiving and storing cells derived from data transmissions conveyed on a network (FIG. 1, dev.12), and for then forwarding the cells elsewhere in the network, wherein each cell is identified as belonging to one of a plurality of flows (para. 31, lines 1-3), wherein each flow has defined minimum and maximum forwarding rates (para. 31, lines 3-7), and wherein each flow is assigned to one of a plurality of virtual output queues (VOQs) (paragraphs 33-36), the method comprising the steps of;

a. sequentially receiving and writing the cells into separate blocks of a cell memory, wherein each memory block is identified by a unique identification number (BLOCK_ID) (FIG. 3, dev. 32) (para. 30, lines 4-10);

b. for each flow for which cells identified as belonging thereto are currently stored in the cell memory, generating BLOCK_IDs of memory blocks storing such cells at a rate bounded by that flow's defined minimum and maximum forwarding rates(FIG. 3, dev. 36) (para. 40);

c. establishing a separate BLOCK_ID queue (FIG. 3, dev. 40/41) corresponding to each of the VOQs, (para. 35)

d. adding each BLOCK_ID generated at step b to a BLOCK_ID queue corresponding to the VOQ to which is assigned the flow of a cell stored in a memory block identified by the generated BLOCK_ID (para. 40),

e. for each VOQ, providing corresponding first back pressure data indicating whether BLOCK_IDs may or may not be removed from the BLOCK_ID queue corresponding to the VOQ (para. 39);

f. for each VOQ, whenever the corresponding first back pressure data indicates BLOCK_IDs may be removed from the corresponding BLOCK_ID queue, successively removing BLOCK_IDs from the corresponding BLOCK_ID queue in an order in which they were added to the BLOCK_ID queue (para. 39);

g. for each VOQ, whenever the corresponding first back pressure data indicates BLOCK_IDs may be not be removed from the corresponding BLOCK_ID queue, refraining from removing BLOCK_IDs from the corresponding BLOCK_ID queue (para. 39); and

h. reading a cell out of the cell memory whenever the BLOCK_ID of the memory block in which the cell is stored is removed from any BLOCK_ID queue at step f (para. 36, lines 1-7).

GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL

Grounds for rejection to be reviewed on appeal are:

whether claims 1 and 2 should be rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent 6,735,203 (HEIMAN),

whether claims 3, 6 and 7 should be rejected fewer than 35 U.S.C. 103(a) as being unpatentable over Heiman in view of U.S. Patent 6,011,779 (WILLS),

whether claims 4 should be rejected under 35 U.S.C. 103(a) as being unpatentable over HEIMAN and WILLS in further view of U.S. Patent 5,689,500 (CHIUSSI), and

whether claims 9 and 10 should be rejected under 35 U.S.C. 103(a) as being unpatentable over HEIMAN in view of U.S. Patent 6,324,165 (FAN).

ARGUMENT

1. Arguments against rejection of claims 1 and 2 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent 6,735,203 (HEIMAN).

HEIMAN teaches a matrix switch for routing sequences of cell between its input and output ports. Each incoming cell includes a header identifying its network source and its destination, and as the switch routes each cell sequences to an appropriate output port, it

translates this information into an output connection identifier ECI_n that identifies the n^{th} sequence to which the cell belongs. HEIMAN's FIG. 5 shows an output port containing a table RST1 for storing cells until they are ready to be forwarded. Table RST1 stores cells of each sequence ECI_n in a separate row. Each m^{th} cell of a given sequence is identified by a separate sequence number SN_m, and a table T2 stores the sequence number of the next cell of each sequence ECI_n to be forwarded. The SN_m number of each cell is included in the incoming cell and check unit CU checks the SN_m number of each incoming cell (CELLS IN) of each sequence ECI_n to determine whether it matches the SN_n number in table T2 for that sequence. If there is a match, the check unit immediately forwards the cell outward via FIFO buffer OB and increments the SN_n number in table T2 for that sequence. Otherwise, the check unit stores the incoming cell in table RST1. See col. 7, lies 16-42. All cells of the same sequence ECI_n are stored in the same row of memory blocks in table RST1. Read unit SU scans the cells in each row of cell memory RST1 to determine whether the SN_m number of any stored cell matches that in table T2, and if so, immediately forwards the cell outward via FIGO buffer OB and increments the SN_m number in table T2 for that sequence. See col. 7, lines 43-65.

Claims 1 and 2

Claim 1 recites "a cell memory for sequentially receiving and storing the incoming cells in separate memory blocks, each of the memory blocks being identified by a unique identification number (BLOCK_ID)." The Examiner correctly cites HEIMAN as teaching a table RST1 or a memory for storing cells, however the Examiner incorrectly assumes that HEIMAN's serial numbers SN_m identify blocks within table RST1 that store the cells. HEIMAN (col. 6, lines 13-26) teaches that each serial number SN_m is a part of the cell and identifies its position within a cell sequence (flow). Claim 1 further recites that the cell memory is also "for thereafter reading out a cell stored in any one of the memory blocks when its BLOCK_ID is transmitted to the cell memory." The Examiner incorrectly reasons that HEIMAN (col. 9, lines 45-67) teaches that table RST1 reads out a cell when a serial number SN_m is transmitted to table RST1 from table T2. Table RST1 does not receive such serial numbers. Check unit CU reads a serial number SN_m out of table T2 only to determine whether it matches a serial

number of an incoming cell, so that it can decide whether to forward the cell or store it in any available location in table RST1.

Claim 1 further recites "queuing means (36) for sequentially generating BLOCK_IDs of memory blocks storing cells to be read out of the cell memory." The Examiner cites HEIMAN (col. 7, lines 15-41) as teaching that control unit CU (FIG. 5) is similar to the queuing means because it generates a sequence of serial numbers SN_m. However a serial number SN_m.which identifies the position of a cell in a sequence, is not the same thing as a BLOCK_ID which identifies a block of memory that stores a cell.

Claim 1 further recites "memory control means (30) for maintaining a BLOCK_ID queue, for adding BLOCK_IDs generated by the queuing means to the BLOCK_ID queue in an order in which they are generated by the queuing means, and for removing BLOCK_IDs from the BLOCK_ID queue and transmitting them to the cell memory in an order in which the BLOCK_IDs were added to the BLOCK_ID queue such that the cell memory reads out the cells stored in the memory blocks identified by the BLOCK_IDs transmitted by the memory control means." The Examiner incorrectly points to HEIMAN (FIG. 5, and col. 6, line 11 to col. 8, line 20 and col. 9, line 46 to col. 10 line 9) as teaching that HEIMAN's serial number unit PSN of FIG. 5, which maintains table T2 storing "order stamps" SN_m anticipates the recited "memory control means" which maintains a BLOCK_ID queue for storing BLOCK_IDs. As discussed above, a serial number SN_m identifies the position of a cell in a sequence and is not the same thing as a BLOCK_ID which identifies a block of memory that stores a cell. Thus table T2 is not a BLOCK_ID queue, since it does not store anything that acts as a BLOCK_ID. Claim 1 recites that the memory control means adds and removes BLOCK_IDs to and from a BLOCK_ID queue (queue 40 or 41 of applicant's FIG. 3), and that each BLOCK_ID identifies a memory block. Thus the applicant's BLOCK_ID queue (40 or 41) keeps track of the BLOCK_IDs of memory blocks within cell memory 32 that contain cells. HEIMAN (col. 6, lines 13-15) teaches that serial number unit PSN of FIG. 5 writes and read serial numbers ("order stamps") SN_m to and from table T2, and that each order stamp SN_m indicates a cell's position within a cell sequence (or flow).

Thus while the memory control means recited in claim 1 maintains a queue storing BLOCK_IDs identifying memory blocks, HEIMAN's serial

number unit PSN of FIG. 5 maintains a table T2 storing serial numbers SN_m identifying individual cells of a cell sequence. Thus HEIMAN fails to teach or suggest the recited "memory control means" of claim 1.

Claim 2 depends on claim 1 and is patentable over HEIMAN for similar reasons.

2. Arguments against rejection of claims 3, 6 and 7 under 35 U.S.C. 103(a) as being unpatentable over HEIMAN in view of U.S. Patent 6,011,779 (WILLS).

Claim 3

With respect to claim 3, the Examiner relies on HEIMAN as anticipating the underlying subject matter of its parent claims 1 and 2 and relies on WILLS as teaching the additional limitations of claim 3. Since, as discussed above in argument 1, HEIMAN fails to teach or suggest the memory control means claims 1 and 2, and since WILLS does not teach such memory control means, claim 3 is patentable over the combination of HEIMAN and WILLS.

Claim 3 recites "the memory control means refrains from removing BLOCK_IDs from the BLOCK_ID queue and transmitting them to the cell memory whenever the first back pressure data indicates that the number of cells stored in the first buffer means is below the first threshold level." The Examiner concedes the HEIMAN teaches nothing about backpressure data indicating whether BLOCK_IDs may be removed from a queue, but cites WILLS (FIG. 6 and col. 6, lines 31-35) as teaching this. WILLS (FIG. 6) shows a network routing system that forward cells from a source buffer 21 to a destination buffer 22 via a sequence of switching elements 11. WILLS (FIG. 6, lines 5-35) teaches that when a cell destination buffer 22 contains a sufficient number of cells, it sends a backpressure (overfill) signal via switching elements 11 to the source buffer 21, telling it to stop sending cells. Source buffer 21 resumes sending cells when buffer 22 turns off the backpressure signal. WILLS does not teach that source buffer 21 forwards cells in memory blocks identified by BLOCK_IDs read out of a queue and that the backpressure signal prevents BLOCK_IDs from being read out of such a queue as recited in claim 3. Thus HEIMAN and WILLS fail to teach or suggest the additional subject matter of claim 3.

Claims 6 and 7

Claim 6 recites a step a of "sequentially receiving and writing the cells into separate blocks of a cell memory, wherein each memory block is identified by a unique identification number (BLOCK_ID)." The Examiner does not cite WILLS as teaching the subject matter of claim 6, step a. but cites HEIMAN (col. 7, lines 3-40) as teaching serial numbers SN_m as being similar in nature to the recited BLOCK_IDs. However, while a BLOCK_ID uniquely identifies a memory block, each of HEIMAN's serial numbers SN_m is an order stamp identifying the position of a cell within a cell sequence. See HEIMAN col. 6, lines 11-18. The Examiner does not cite WILLS as teaching the subject matter of claim 6, step a. Thus HEIMAN and WILLS fail to teach or suggest the subject matter of claim 6, step c.

Claim 6 recites a step b of "generating a sequence of BLOCK_IDs of memory blocks currently storing cells." The Examiner does not cite WILLS as teaching the subject matter of claim 6, step b but cites HEIMAN (col. 7, lines 15-41) as teaching a check unit (FIG. 5, CU) that generates a sequence of serial numbers SN_m and a sequence of "connection identifiers" ECI_n. The examiner apparently argues that either the SN_m or ECI_n sequence is equivalent to the recited BLOCK_ID sequence. However as discussed above, ECI_n numbers identify cell sequences (or "flows") and corresponding rows of memory blocks, and SN_m numbers identify positions of cells within a cell sequence. Since the recited BLOCK_IDs identify individual memory blocks that store cells, generating a sequence of SN_m or ECI_n numbers is not equivalent to generating a sequence of BLOCK_IDs as recited in claim 6, step b. Thus HEIMAN and WILLS fail to teach or suggest the subject matter of claim 6, step b.

Claim 6 recites a step c of "adding each generated BLOCK_ID to a BLOCK_ID queue." The Examiner cites HEIMAN's table T2 of FIG. 5 as being equivalent to the recited block ID queue. However, as seen in HEIMAN's FIG. 5, table T2 stores serial numbers SN_m, each identifying a particular cell of a sequence, and does not store BLOCK_IDs identifying memory blocks as recited in claim 6, step c. The Examiner does not cite WILLS as teaching the subject matter of claim 6, step c. Thus HEIMAN and WILLS fail to teach or suggest the subject matter of claim 6, step c.

Claim 6 recites a step d of successively removing BLOCK_IDs from the BLOCK_ID queue in an order in which BLOCK_IDs were added to the BLOCK_ID queue whenever the BLOCK_ID queue contains BLOCK_IDs and first back pressure data indicates that BLOCK_IDs may be removed from the BLOCK_ID queue, and refraining from removing BLOCK_IDs from the BLOCK_ID queue when the first back pressure data indicates that BLOCK_IDs may not be removed from the BLOCK_ID queue. The Examiner cites HEIMAN as teaching that removing SN_m numbers for table T2 of FIG. 5 is equivalent to removing BLOCK_IDs from a BLOCK_ID queue as recited in claim 6, step d. However, as discussed above, HEIMAN's SN_m numbers are not equivalent to the recited BLOCK_IDs. Also those of skill in the art will appreciate that a table T and a queue are not the same thing. The Examiner concedes the HEIMAN teaches nothing about backpressure data indicating whether BLOCK_IDs may be removed from a queue, but cites WILLS (FIG. 6 and col. 6, lines 31-35) as teaching this. WILLS (FIG. 6) shows a network routing system that forward cells from a source buffer 21 to a destination buffer 22 via a sequence of switching elements 11. WILLS (FIG. 6, lines 5-35) teaches that when cell destination buffer 22 contains a sufficient number of cells, it sends a backpressure (overfill) signal via switching elements 11 back to the source buffer 21, telling it to stop sending cells. Source buffer 21 resumes sending cells when destination buffer 22 turns off the backpressure signal. WILLS does not teach that source buffer 21 forwards cells in memory blocks identified by BLOCK_IDs read out of a queue and that the backpressure signal prevents BLOCK_IDs from being read out of such a queue as recited in claim 6, step d. Thus HEIMAN and WILLS fail to teach or suggest the subject matter of claim 6, step d.

Claim 6 further recites a step e of "reading a cell out of the cell memory whenever the BLOCK_ID of the memory block in which it is stored is removed from the BLOCK_ID queue at step d." The Examiner incorrectly cites HEIMAN (FIG. 5, col. 9, lines 45-67) as teaching the subject matter of steps e, stating that cells are read out of table RST1 when the "serial number [SN_m] of the each cell is transmitted to the RST1 memory." As discussed above, the serial numbers SN_m stored in table T2 are not BLOCK_IDs identifying blocks of table RST1 containing cells; they identify the next cell of a sequence to be forwarded, and that cell may or may not currently reside in table RST1. Also table

RST1 does not read out cells when the serial number SN_m of a cell is read out of table T2. Serial numbers SN_m read out of table T2 are not transmitted to table RST1; they are compared to serial numbers of incoming cells to determine whether the cells should be forwarded. The Examiner does not cite WILLS as teaching the subject matter of claim 6, step e. Thus HEIMAN and WILLS fail to teach or suggest the subject matter of claim 6, step e.

Claim 7 depends on claim 6 and is patentable over the combination of HEIMAN and WILLS for similar reasons

3. Arguments against rejection of claim 4 under 35 U.S.C. 103(a) as being unpatentable over HEIMAN and WILLS in further view of U.S. Patent 5,689,500 (CHIUSSI).

Claim 4

Since the Examiner relies on the combination of HEIMAN and WILLS as anticipating the subject matter of the parent claim 3 of claim 4, claim 4 is also patentable over the combination of HEIMAN and WILLS for the reasons set forth above in connection with claim 3. Since the Examiner relies on CHIUSSI only as teaching the additional limitations of claim 4 and not as teaching any limitations of claim 3, claim 4 is patentable over the combination of HEIMAN, WILLS and CHIUSSI.

4. Arguments against rejection of claims 9 and 10 under 35 U.S.C. 103(a) as being unpatentable over HEIMAN in view of U.S. Patent 6,324,165 (FAN).

Claims 9 and 10

The invention as recited in claim 9 is a method for receiving and storing cells derived from data transmissions conveyed on a network (FIG. 1, dev.12), and for then forwarding the cells elsewhere in the network, wherein each cell is identified as belonging to one of a plurality of flows (para. 31, lines 1-3), wherein each flow has defined minimum and maximum forwarding rates (para. 31, lines 3-7), and wherein each flow is assigned to one of a plurality of virtual output queues (VOQs) (paragraphs 33-36). The Examiner correctly cites HEIMAN as teaching that cells can be parts of different flows (cell

sequences or "connections"), and that such cells can be temporarily stored in a table RST1 (FIG. 5).

Claim 9 recites a step a "sequentially receiving and writing the cells into separate blocks of a cell memory, wherein each memory block is identified by a unique identification number (BLOCK_ID)." As discussed above, the Examiner incorrectly cites HEIMAN's serial numbers SN_m indicating the position of a cell within a cell sequence (flow) as equivalent to the recited BLOCK_ID referencing a memory block storing a cell.

Claim 9 recites a step (b) of "for each flow for which cells identified as belonging thereto are currently stored in the cell memory, generating BLOCK_IDs of memory blocks storing such cells at a rate bounded by that flow's defined minimum and maximum forwarding rates." The Examiner concedes HEIMAN does not teach step b but cites FAN (col. 6, line 20- col. 7, line 6) as teaching to generate BLOCK_IDs at a rate bounded by a flows minimum and maximum forwarding rates. However FAN's system does not include a cell memory having blocks for storing cells at locations indicated by BLOCK_IDs, and therefore does not generate BLOCK_IDs at any rate. Thus HEIMAN and FAN fail to teach or suggest the subject matter of claim 9, step b.

Claim 9 recites a step (c) of "establishing a separate BLOCK_ID queue corresponding to each of the VOQs" where each flow is assigned to one of a plurality of virtual output queues (VOQs). The Examiner does not cite FAN as teaching establishing a BLOCK_ID queue, but reasons that HEIMAN's table T2 is equivalent to a BLOCK_ID queue. However as discussed above, table T2 stores SN_m numbers identifying particular cells within flows and does not store BLOCK_IDs identifying memory blocks storing cells. HEIMAN's table T2 is therefore not equivalent to the recited BLOCK_ID queues. Thus HEIMAN and FAN fail to teach or suggest the subject matter of claim 9, step c.

Claim 9 recites a step d of "adding each BLOCK_ID generated at step b to a BLOCK_ID queue corresponding to the VOQ to which is assigned the flow of a cell stored in a memory block identified by the generated BLOCK." The Examiner reasons that HEIMAN's table T2 is equivalent to a BLOCK_ID queue; however, as discussed above, table T2 is not a queue, and the SN_m numbers it stores are not equivalent to BLOCK_IDs. The Examiner also cites FAN (col. 6, line 56 through col. 7, line 9) as teaching to add BLOCK_IDs to a BLOCK_ID queue, however

since FAN's system does not include a cell memory having blocks for storing cells at locations indicated by BLOCK_IDs, FAN's system does not add BLOCK_IDs to a BLOCK_ID queue. Thus HEIMAN and FAN fail to teach or suggest the subject matter of claim 9, step d.

Claim 9 recites a step e of "for each VOQ, providing corresponding first back pressure data indicating whether BLOCK_IDs may or may not be removed from the BLOCK_ID queue corresponding to the VOQ." The Examiner concedes HEIMAN does not teach this, but cites FAN as teaching VOQs though not as disclosing BLOCK_ID queues. However, since FAN's system does not include a cell memory for storing cells at locations indicated by BLOCK_IDs, it does not maintain a BLOCK_ID queue. Thus HEIMAN and FAN fail to teach or suggest the subject matter of claim 9, step e.

Claim 9 recites a step f of "for each VOQ, whenever the corresponding first back pressure data indicates BLOCK_IDs may be removed from the corresponding BLOCK_ID queue, successively removing BLOCK_IDs from the corresponding BLOCK_ID queue in an order in which they were added to the BLOCK_ID queue." The examiner cites FAN (Col. 8, lines 9-67) as teaching this. However, FAN/s backpressure signal STOP NRT does not indicate when BLOCK-IDs may be removed from a BLOCK_ID queue since FAN's system does not maintain a BLOCK_ID queue. Thus HEIMAN and FAN fail to teach or suggest the subject matter of claim 9, step f.

Claim 9 recites a step g of "for each VOQ, whenever the corresponding first back pressure data indicates BLOCK_IDs may be not be removed from the corresponding BLOCK_ID queue, refraining from removing BLOCK_IDs from the corresponding BLOCK_ID queue." The Examiner concedes HEIMAN does not teach this but cites FAN (col. 8, lines 9-67) as teaching step g. However, FAN's system does not include a cell memory for storing cells at locations indicated by block IDs, and has no BLOCK_ID queue, it has no need to refrain from removing BLOCK_IDs from any BLOCK_ID queue in response to backpressure data. Thus HEIMAN and FAN fail to teach or suggest the subject matter of claim 9, step g.

Claim 9 recites a step h of "reading a cell out of the cell memory whenever the BLOCK_ID of the memory block in which the cell is stored is removed from any BLOCK_ID queue at step f." The Examiner does not cite FAN as teaching the subject matter of claim 9, step h,

but cites HEIMAN (FIG. 5, col. 9, lines 45-67) as teaching that cells are read out of table RST1 when the "serial number [SN_m] of the each cell is transmitted to the RST1 memory." As discussed above, the serial numbers SN_m stored in table T2 are not BLOCK_IDs identifying blocks of table RST1 containing cells; they identify the next cell of a flow (cell sequence) to be forwarded, and that particular cell may or may not currently reside in table RST1. Also table RST1 does not read out cells when the serial number SN_m of a cell is read out of table T2 is transmitted to table RST1. As discussed above, serial numbers SN_m read out of table T2 are not transmitted to table RST1; they are compared to serial numbers of incoming cells to determine whether the cells should be forwarded or stored in table RST1. Thus HEIMAN and FAN fail to teach or suggest the subject matter of claim 9, step g.

Claim 10 is patentable over the combination of HEIMAN and FAN for reasons set forth above in connection with its parent claim 9. Claim further 10 recites a step k of "for each VOQ, generating the corresponding first back pressure data, wherein the first back pressure data indicates that BLOCK_IDs may be removed from the BLOCK_ID queue corresponding to the VOQ when a number of cells of flows assigned to that VOQ stored in the first buffer means is below a first threshold level, and wherein the first back pressure data indicates that BLOCK_IDs may not be removed from the BLOCK_ID queue corresponding to the VOQ when a number of cells of flows assigned to that VOQ stored in the first buffer means is above the first threshold level." The Examiner cites HEIMAN (col. 8, lines 1-67), but not FAN, as teaching to read BLOCK_IDs out of a BLOCK_ID queue and teaches FAN as teaching backpressure signals. However it would not be obvious to use FAN's backpressure signal to indicate that BLOCK_IDs may be removed from a HEIMAN's BLOCK_ID queue since, as discussed above, HEIMAN does not teach to maintain a BLOCK_ID queue. The Examiner confuses HEIMAN's table T2 with a BLOCK_ID queue. Thus HEIMAN and FAN fail to teach or suggest the subject matter of claim 10, step k.

CLAIMS APPENDIX

1. An apparatus for receiving and storing incoming cells derived from data transmissions conveyed on a network, and for then forwarding the cells elsewhere in the network, the apparatus comprising:

a cell memory for sequentially receiving and storing the incoming cells in separate memory blocks, each of the memory blocks being identified by a unique identification number (BLOCK_ID), and for thereafter reading out a cell stored in any one of the memory blocks when its BLOCK_ID is transmitted to the cell memory;

queuing means (36) for sequentially generating BLOCK_IDs of memory blocks storing cells to be read out of the cell memory; and

memory control means (30) for maintaining a BLOCK_ID queue, for adding BLOCK_IDs generated by the queuing means to the BLOCK_ID queue in an order in which they are generated by the queuing means, and for removing BLOCK_IDs from the BLOCK_ID queue and transmitting them to the cell memory in an order in which the BLOCK_IDs were added to the BLOCK_ID queue such that the cell memory reads out the cells stored in the memory blocks identified by the BLOCK_IDs transmitted by the memory control means.

2. The apparatus in accordance with claim 1 further comprising:

first buffer means (37) for storing cells read out of the cell memory, and for thereafter reading out and forwarding cells it has stored.

3. The apparatus in accordance with claim 2

wherein the first buffer means produces and sends first back pressure data to the memory control means indicating whether a number

of cells stored in the first buffer means is above a first threshold level,

wherein the memory control means successively removes BLOCK_IDs from the BLOCK_ID queue and transmits them to the cell memory whenever the BLOCK_ID queue contains at least one BLOCK_ID and the first back pressure data indicates that the number of cells stored in the first FIFO buffer means is above the first threshold level, and

wherein the memory control means refrains from removing BLOCK_IDs from the BLOCK_ID queue and transmitting them to the cell memory whenever the first back pressure data indicates that the number of cells stored in the first buffer means is below the first threshold level.

4. The apparatus in accordance with claim 3 further comprising

second buffer means (24) for storing cells read out of the first buffer means and for thereafter forwarding each cell it stores elsewhere in the network.

5. The apparatus in accordance with claim 4

wherein the second buffer means generates second back pressure data indicating an amount of unoccupied storage capacity of the second buffer means, and

wherein the first buffer means sets the first threshold level in response to the second back pressure data.

6. A method for receiving and storing cells derived from data transmissions conveyed on a network, and for then forwarding the cells elsewhere in the network, the method comprising the steps of;

a. sequentially receiving and writing the cells into separate blocks of a cell memory, wherein each memory block is identified by a unique identification number (BLOCK_ID);

b. generating a sequence of BLOCK_IDs of memory blocks currently storing cells;

c. adding each generated BLOCK_ID to a BLOCK_ID queue,

d. successively removing each BLOCK_ID from the BLOCK_ID queue in an order in which BLOCK_IDs were added to the BLOCK_ID queue whenever the BLOCK_ID queue contains BLOCK_IDs and first back pressure data indicates that BLOCK_IDs may be removed from the BLOCK_ID queue, and refraining from removing BLOCK_IDs from the BLOCK_ID queue when the first back pressure data indicates that BLOCK_IDs may not be removed from the BLOCK_ID queue,

e. reading a cell out of the cell memory whenever the BLOCK_ID of the memory block in which it is stored is removed from the BLOCK_ID queue at step d.

7. The method in accordance with claim 6 further comprising the steps of:

f. writing cells read out of the cell memory into first buffer means for storing and reading out cells,

g. reading the cells out of the first buffer means, and

h. setting the first back pressure data to indicate that BLOCK_IDs may not be removed from the BLOCK_ID queue whenever a number of cells stored in the first buffer means rises above a threshold level, and setting the first back pressure data to indicate that BLOCK_IDs may be removed from the BLOCK_ID queue whenever the number

of cells stored in the first buffer means falls below the threshold level.

8. The method in accordance with claim 7 further comprising the steps of:

i. storing cells read out of the first buffer means in second buffer means for storing and reading out cells;

j. reading cells out of the second buffer means and forwarding them elsewhere in the network;

k. generating a second back pressure data indicating an amount of unoccupied storage capacity of the second buffer means; and

l. setting the threshold level in response to the value of the code conveyed in the second back pressure data.

9. A method for receiving and storing cells derived from data transmissions conveyed on a network, and for then forwarding the cells elsewhere in the network, wherein each cell is identified as belonging to one of a plurality of flows, wherein each flow has defined minimum and maximum forwarding rates, and wherein each flow is assigned to one of a plurality of virtual output queues (VOQs), the method comprising the steps of;

a. sequentially receiving and writing the cells into separate blocks of a cell memory, wherein each memory block is identified by a unique identification number (BLOCK_ID);

b. for each flow for which cells identified as belonging thereto are currently stored in the cell memory, generating BLOCK_IDs of

memory blocks storing such cells at a rate bounded by that flow's defined minimum and maximum forwarding rates;

c. establishing a separate BLOCK_ID queue corresponding to each of the VOQs,

d. adding each BLOCK_ID generated at step b to a BLOCK_ID queue corresponding to the VOQ to which is assigned the flow of a cell stored in a memory block identified by the generated BLOCK_ID,

e. for each VOQ providing corresponding first back pressure data indicating whether BLOCK_IDs may or may not be removed from the BLOCK_ID queue corresponding to the VOQ;

f. for each VOQ, whenever the corresponding first back pressure data indicates BLOCK_IDs may be removed from the corresponding BLOCK_ID queue, successively removing BLOCK_IDs from the corresponding BLOCK_ID queue in an order in which they were added to the BLOCK_ID queue;

g. for each VOQ, whenever the corresponding first back pressure data indicates BLOCK_IDs may be not be removed from the corresponding BLOCK_ID queue, refraining from removing BLOCK_IDs from the corresponding BLOCK_ID queue; and

h. reading a cell out of the cell memory whenever the BLOCK_ID of the memory block in which the cell is stored is removed from any BLOCK_ID queue at step f.

10. The method in accordance with claim 9 further comprising the steps of:

i. storing each cell read out of the cell memory in first buffer means,

j. reading the cells out of the first buffer means, and

k. for each VOQ, generating the corresponding first back pressure data,

wherein the first back pressure data indicates that BLOCK_IDs may be removed from the BLOCK_ID queue corresponding to the VOQ when a number of cells of flows assigned to that VOQ stored in the first buffer means is below a first threshold level, and

wherein the first back pressure data indicates that BLOCK_IDs may no be removed from the BLOCK_ID queue corresponding to the VOQ when a number of cells of flows assigned to that VOQ stored in the first buffer means is above the first threshold level.

11. The method in accordance with claim 10 further comprising the steps of:

l. storing cells read out of the first buffer means in second buffer means;

m. reading cells out of the second buffer means and forwarding them elsewhere in said network;

n. for each VOQ, generating corresponding second back pressure data indicating an amount of unoccupied storage capacity of the second buffer means, and

o. adjusting the first threshold level corresponding to each VOQ in response to the second back pressure data corresponding to that VOQ.

12. The method in accordance with claim 11 further comprising the step of:

p. for each VOQ, generating a corresponding third back pressure data indicating whether a number of BLOCK_IDs residing in the

corresponding BLOCK_ID queue is above or below a second threshold level,

wherein a rate at which BLOCK_IDs of cells identified as belonging to each flow assigned to the VOQ are generated at step g is set to the flow's assigned minimum forwarding rate when the corresponding third back pressure data indicates that the number of BLOCK_IDs residing in the corresponding BLOCK_ID queue is above the second threshold level, and

wherein the rate at which BLOCK_IDs of cells identified as belonging to each flow assigned to the VOQ are generated at step g is set higher than the flow's assigned minimum forwarding rate when the corresponding third back pressure data indicates that the number of BLOCK_IDs residing in the corresponding BLOCK_ID queue is below the second threshold level.

13. The method in accordance with claim 9 further comprising the step of:

i. for each VOQ, generating a corresponding third back pressure data indicating whether a number of BLOCK_IDs residing in the corresponding BLOCK_ID queue is above or below a second threshold level,

wherein a rate at which BLOCK_IDs of cells identified as belonging to each flow assigned to the VOQ are generated at step g is set to the flow's assigned minimum forwarding rate when the corresponding third back pressure data indicates that the number of BLOCK_IDs residing in the corresponding BLOCK_ID queue is above the second threshold level, and

wherein the rate at which BLOCK_IDs of cells identified as belonging to each flow assigned to the VOQ are generated at step g is set higher than the flow's assigned minimum forwarding rate when the corresponding third back pressure data indicates that the number of BLOCK_IDs residing in the corresponding BLOCK_ID queue is below the second threshold level.

EVIDENCE APPENDIX

Not applicable.

RELATED PROCEEDINGS APPENDIX

Not Applicable.

Respectfully submitted,

Daniel J. Bedell
Reg. No. 30,156

SMITH-HILL & BEDELL, P.C.
16100 N.W. Cornell Road, Suite 220
Beaverton, Oregon 97006

Tel. (503) 574-3100
Fax (503) 574-3197
Docket: ZETT 2151
Postcard: 11/05-18

Certificate of Mailing

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on the 10 day of November, 2005.

Penelope Stockwell